Criando o Personagem e suas configurações

Muito bem pessoal. Agora que já sabemos como usar a programação C# dentro da unity, estamos preparados para criar nosso game e seu backend, ou seja, a programação que ficará por trás do nosso game.

Antes de qualquer coisa, é super importante definir qual é o objetivo do jogo, pois a partir dele que você irá definir também os assets a serem utilizados no game como personagem, o cenário, quais serão os objetos de interação com o cenário e com o personagem, etc, etc.

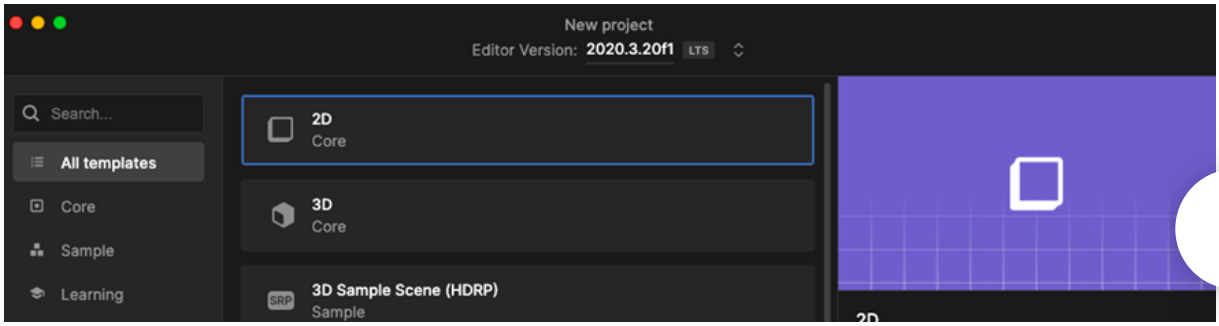
Além disso, é fundamental ter claro qual a mecânica do seu jogo, ou seja, quais serão as regras estabelecidas para o jogo funcionar.

E claro, um nome bem atrativo para o seu game! Isso irá chamar a atenção para as pessoas baixarem e jogarem seu game.

Tendo isso pronto, vamos iniciar a construção do nosso game!!

O jogo que criaremos como exemplo irá se chamar AsteroidX!

Abra o Unity Hub, clique em New Project:



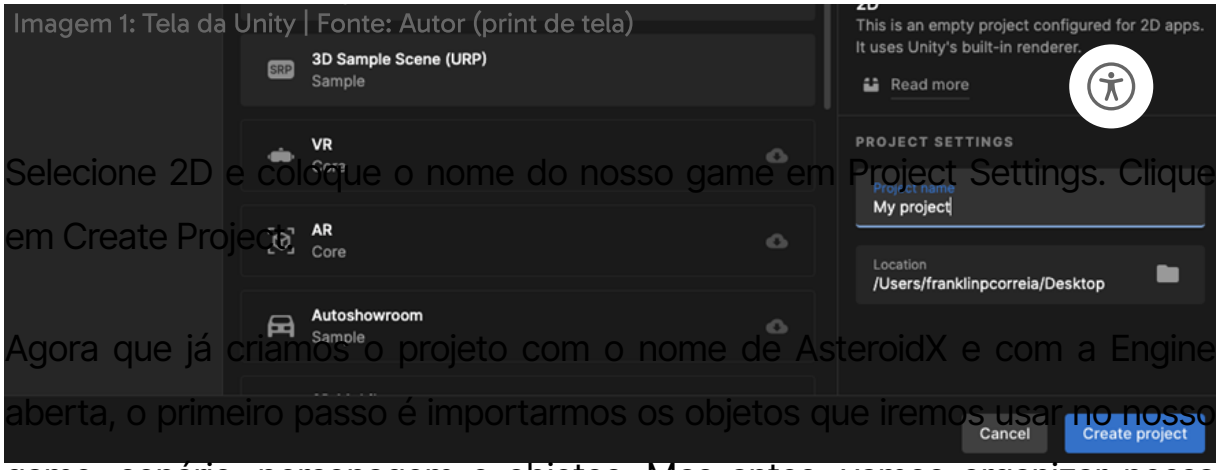


Imagem 1: Tela da Unity | Fonte: Autor (print de tela)

Selecione 2D e coloque o nome do nosso game em Project Settings. Clique em Create Project.

Agora que já criamos o projeto com o nome de AsteroidX e com a Engine aberta, o primeiro passo é importarmos os objetos que iremos usar no nosso game: cenário, personagem e objetos. Mas antes, vamos organizar nossa área de trabalho.

Crie duas pastas no painel Projects com o nome de Objects2D e uma outra pasta chamada Scripts para inserirmos toda a nossa programação C#:

Para importar, podemos fazer de várias formas:

Você pode arrastar direto as imagens da sua pasta para a pasta Assets – Objects2D, ou simplesmente clicar com o botão da direita do mouse dentro da pasta Objects2D e clicar em Import New Assets.

Escolha a melhor forma e importe os arquivos.

Muito bem. Já temos agora nossos objetos dentro da engine. Agora, vamos organizar nosso cenário e entender como posicioná-los no cenário.

Arraste para o painel hierarchy o fundo do cenário e faça com que ele fique um pouco maior que o raio da nossa câmera. Você pode usar as teclas Shift + Alt para aumentar ou diminuir proporcional.

Agora, arraste a nossa nave, que será o personagem principal do nosso game. Também posicione a nave no cenário em um posicionamento adequado para seu uso dentro do game:

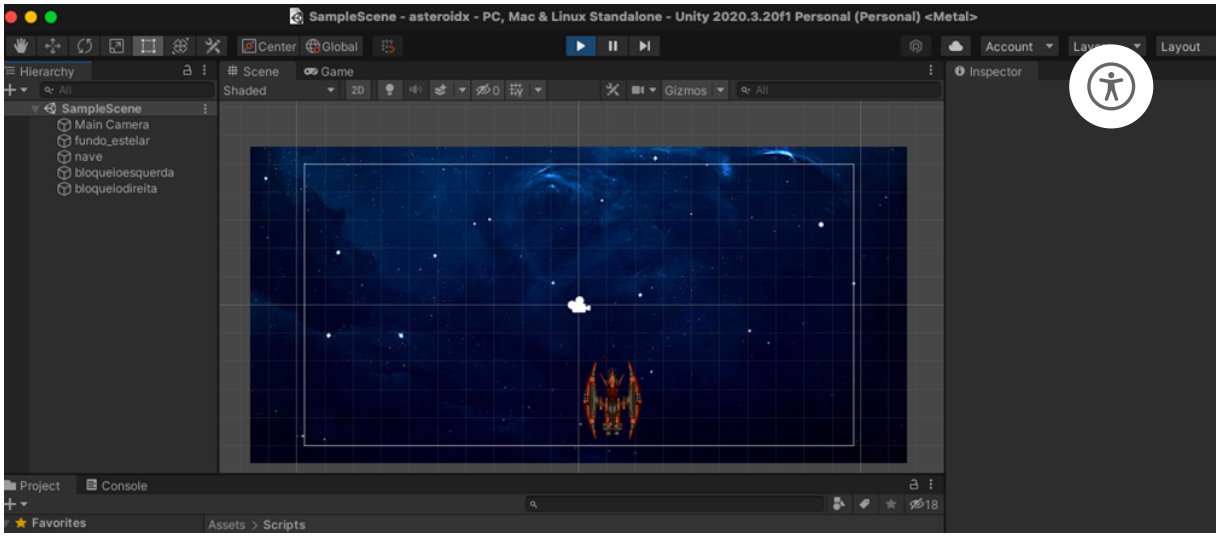


Imagem 2: Tela da Unity | Fonte: Autor (print de tela)

Para realizar a programação da nossa nave conforme o passo a passo na nossa vídeo-aula, lembre-se que é importante atribuir a velocidade adequada da nave para que a funcionalidade de movimentação dela não fique muito lenta, ou muito rápida, aumentando a dificuldade do jogo ou deixando o jogo muito fácil.

Abaixo temos a programação em C# que usamos para este módulo:

Criação do método para mover a nave:

```
public void movenave()
  {
                                        float cmdteclado =
Input.GetAxisRaw("Horizontal") velocidade Time.deltaTime;
transform.position = new Vector3((transform.position.x + cmdteclado),
transform.position.y, transform.position.z);
```

```
}
```

Após inserir toda a programação, realize os testes no jogo para verificar seu funcionamento. Não esqueça de salvar seu projeto sempre, clicando em File – Save.

Atividade Extra

Acesse o manual de componentes da Unity no site https://docs.unity3d.com/Manual/UnityManual.html e conheça detalhadamente todos os recursos e componentes. Neste manual, constam todos os componentes com suas explicações e exemplos para testar na Unity. A maioria das explicações estão em inglês. Use o translate no navegador e coloque em tradução automática para o português.

Realize alguns testes com os exemplos prontos que te chamar mais a atenção.

Referência Bibliográfica

UNITY, Software. Site do Produto.

https://unity.com/pt/

https://docs.microsoft.com/pt-br/dotnet/csharp/

https://docs.unity3d.com/Manual/UnityManual.html

Ir para exercício